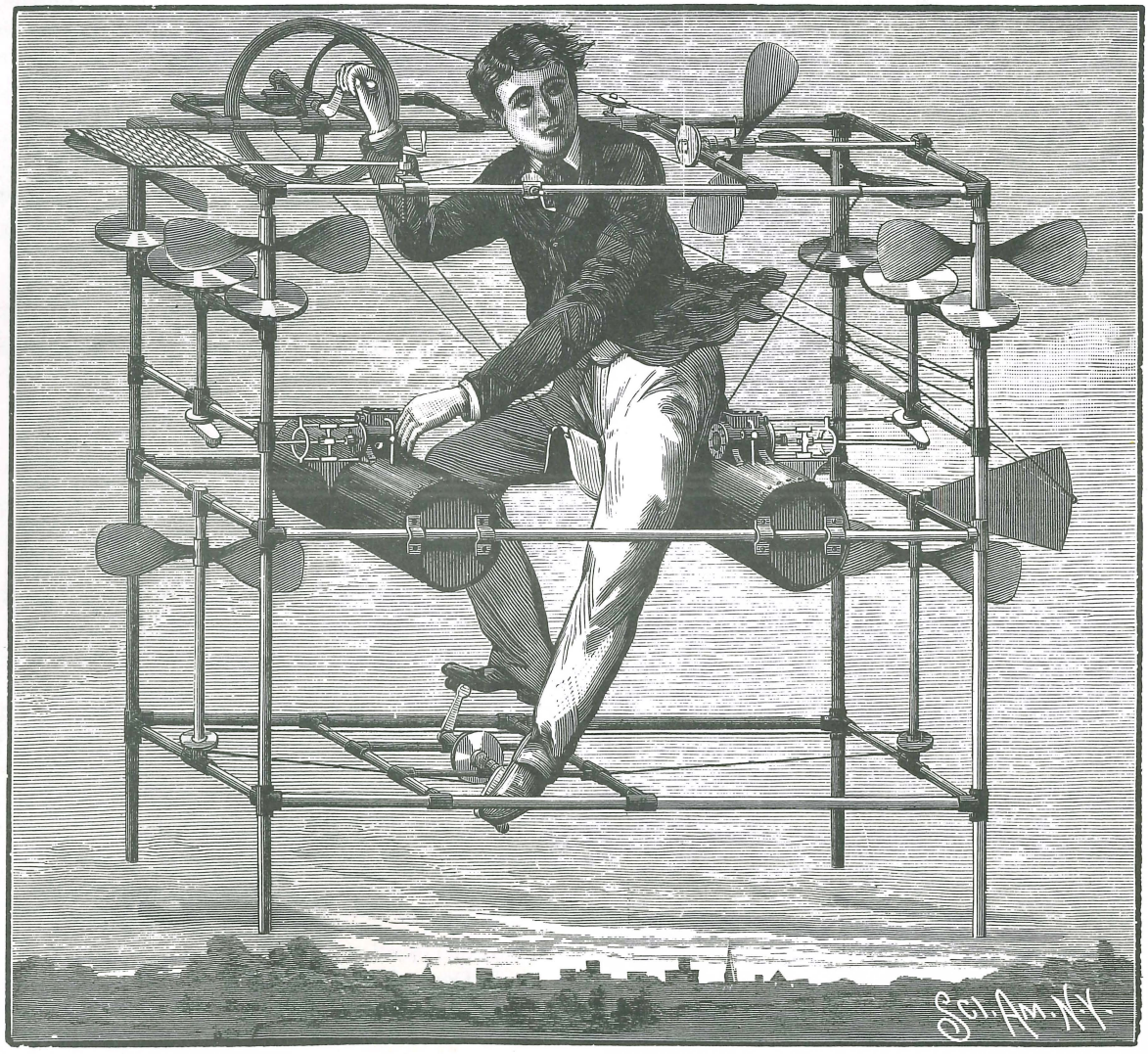
THE SEED

January 1987
Vol. 9 No. 1

NEWSLETTER FOR APPLE PEOPLE

DENVER APPLE PI



denver apple pi Seed

P.O. BOX 17467
DENVER, COLORADO 80217

President .....Trent Hein
Vice President & Property .....
Secretary & Editor .....Randall Ansley
Treasurer .....Bob Hein
Board Member & Software Librarian .....
Board Member & Store Rep .....Tom Angelos
Board Member & Sales Officer .....Chris Saxon
BBS Officer .....Robert Racansky

MEMBERSHIP INFORMATION

APPLE PI is an Apple Computer Users Group - not affiliated with Apple Computer, Inc. - that provides many member benefits, some of which are:

MONTHLY MEETINGS - On the first Thursday of each month at 7:00 p.m. APPLE PI holds its general meeting at the Green Center, Colorado School of Mines Campus, Golden, Colorado. This meeting is a combination of an informational meeting, new product announcements, and a short business meeting. The members are given an opportunity to buy computer supplies a little above cost. Software products are given away as door prizes.

On the Sunday following the general meeting there is an informal meet which is usually held at the Clear Creek Office Park, 4251 Kipling, in Wheat Ridge. This meeting is from 10:00 a.m. until 3:00 p.m. and provides a more informal environment to seek or give advice to fellow members and get copies of programs from the APPLE PI Library. Both of these meetings are open to the general public but only members have access to the club library and products that are sold.

PROGRAM LIBRARY - APPLE PI currently has a library of 9,000 programs. All current disks are in 16 sector (3.3, Pascal, CP/M) format and are available to any member. APPLE PI Members can obtain copies from the library:

1) By coming to the monthly informal meeting (swap meet);
2) By written request to the club librarian (at any meeting or by mail);
3) By electronic mail on APPLE PI's bulletin board system:

DISK OF THE MONTH - Each month a collection of some of the best programs donated to APPLE PI during the preceding month are made available. The DOM can be purchased at any group meeting or by mail.

DUES - Current membership dues are **$18.00** per year, with a one-time application fee of **$10.00.** Checks should be made payable to DENVER APPLE PI.

The words APPLE, APPLE COMPUTER, and the APPLE LOGO are registered trademarks of the Apple Computer Company Inc. Representation of these trademarks in the Seed does not constitute any expression of agreement or disagreement by Apple Computer Inc. with any articles or other printed material found in the Seed. The opinions, policies and comments contained in the Seed are those of the Apple Pi users group, officers and members.

NOTICE

Articles, stories, and other items in the Seed may be copied or used by other clubs or groups as long as credit is given to Denver Apple Pi and the author of the article. On reprints please include the date or volume number from the Seed.

ADVERTISING RATES

1/4 Page	$ 25.00
1/2 Page	$ 40.00
Full Page	$ 70.00

PRESIDENTS'S REPORT

by T.R. Hein

Happy 1987! All of us at Apple Pi hope you had a terrific holiday season. As you may be aware, the holiday season brought many changes to Apple Pi with the loss of 2 of our officers: our Librarian, Ed Thompson, who job was transferred to another state, and our President, Roy Dean, who will be on a teacher exchange in Australia for a year. Certainly, they were some of the most active officers in Apple Pi, and we will miss both of them VERY much. Of course, we do wish them the best of luck in their new positions.

We did, however, gain a new officer, as Chris Saxon was appointed to fill a vacant Director-at-Large position at the November board meeting. Chris has been serving as our Sales Officer and one of Assistant Librarians. Another member, Chuck Habanicht, will be taking care of our MacIntosh and MacIntosh section of the library, allowing Apple Pi to serve the Mac user much better. We thank Chris and Chuck for volunteering to help us in these vital positions.

Unfortunately, that still leaves some very crucial positions open. Jack Norton, who has been printing the SEED for the last few years, has already volunteered to help with the librarian's duties, but would also like some help doing it. If you're interested in helping out with the fantastic Denver Apple Pi software library, or you'd be interested in another position on the board (perhaps Vice-President?) PLEASE PLEASE PLEASE contact one of the officers as soon as possible!!! We desperately need YOUR HELP!!! All of the people who do all those great things for you at Apple Pi are VOLUNTEERS, but we seem to have a shortage of them!!! This is your chance to become involved and experience the rewards of helping your fellow computer users. We'll be glad to talk about any position your interested in.

Hopefully, all of you were aware that the January general meeting was moved to the 8th because of a conflict the 1st had with New Year's Day. At the January meeting, our speaker will be Mark Robbins from United Software demonstrating MouseTalk. At the February general meeting, Dave Eisler will be demonstrating some fantastic new software that is now on the market. I hope we will see you at both of these meetings. Also, remember that the Core Meets are always held on the Sunday following the general meeting from 10am-3pm at Clear Creek Office Park, 4251 Kipling, in Wheatridge.

Finally, I'd like to encourage you to bring a friend to Apple Pi. Perhaps you have a friend or friends who received a new computer for Christmas or other reason in the recent past. Tell them about us, and bring them to a meeting. We'd love to meet them and I'm sure they'd find something interesting they didn't know at one of our meetings. I look forward to seeing all of you soon.

BBS PHONE NUMBER

422 - 3716

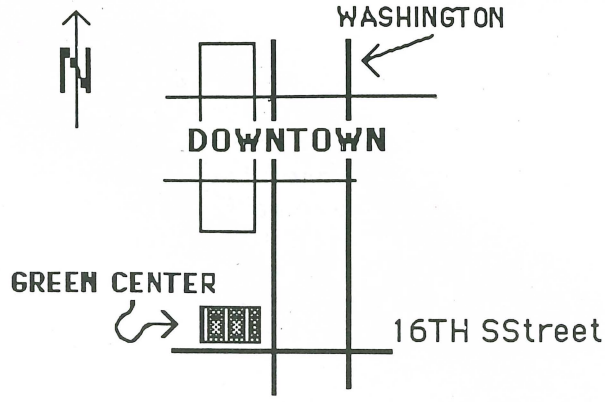
HOT LINE

422 - 0005

TREASURER'S REPORT

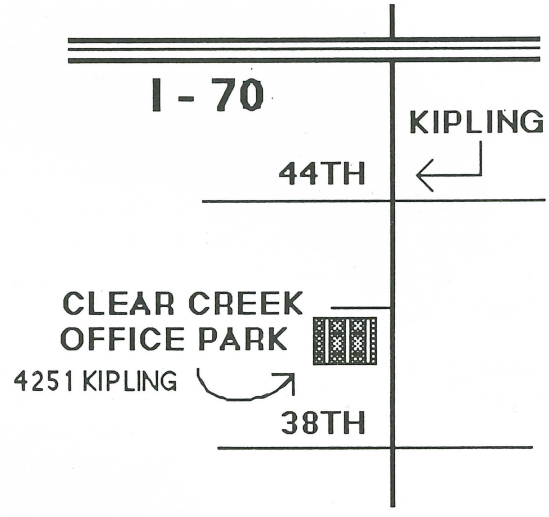
ENDING CASH BALANCE	11/30/86	$1701.28
INCOME		
New Members		140.00
Renewal Members		36.00
DOM Sales		165.00
Library Copies		48.00
Sales of Supplies		638.00
Store Memberships		0.00
Interest Earned		
TOTAL INCOME		$1027.00
EXPENSES		
Supplies for Resale		0.00
Hot Line		0.00
CSM Room Rental		0.00
Printing Costs		0.00
Postage		0.00
Supplies and Copies		0.00
BBS Expense		0.00
State Sales Tax		0.00
Equipment Repairs		0.00
Equipment Rental		0.00
Christmas Party		
TOTAL EXPENSES		$163.45
ENDING CASH BALANCE	12/31/86	
ASSETS	12/31/86	
Cash on Hand		25.00
Cash in Bank		2564.83
Inventory for Resale		1308.50
Prepaid Expenses		365.60
Postage account @ $356.60		
Accounts Receivable		0.00
Equipment (Replacement Cost)		10,850.00
Total Assets		$15,113.93
LIABILITIES		
Sales Tax Accrued		342.48
Total Liabilities		$342.48
NET WORTH OF CLUB as of 12/31/86		$14771.45

GENERAL MEETING

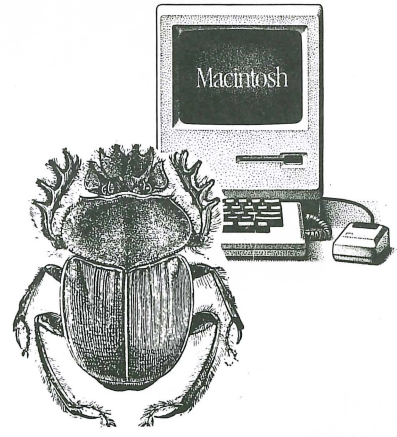


COLO. SCHOOL OF MINES

CORE MEET



COMPUTER BUG



Stories from the past...SEED June 1984

Get ProDOS Ticking...

by Randall Ansley

If you are using ProDOS and you have a Thunderclock (clock card) installed, ProDOS will read the date and time, and use them in any number of routines. The most common use of date and time is in the catalog. When you save a file ProDOS will place the date and time of the first save and the most recent file update in the catalog display for you.

```
/PRODOS

NAME            TYPE  BLOCKS  MODIFIED         CREATED

*PRODOS          SYS     31   1-JAN-84  9:15  <NO DATE>
*BASIC.SYSTEM    SYS     21   15-NOV-84 0:00  <NO DATE>
 STARTUP         BAS      7   15-APR-84 20:15  3-MAR-84  9:07
 LETTER          TXT     13   17-APR-84 9:45  <NO DATE>

BLOCKS FREE  208     BLOCKS USED:  76     TOTAL BLOCKS:  280
```

(sample catalog with "CAT" command)

You can make use or other clock cards but you will need to write a special routine to read your card and place the date and time in the proper locations. The ProDOS Machine Language Interface (MLI) use the following steps:

1. If the date or time field in the call's parameter list is a nonzero, then the ProDOS MLI will use the stored values. If 00 is found then the MLI will call location $BF06. This location may contain a JMP to a clock / calendar routine. The defalt is to a known RTS instruction (this routine may or may not update the system date and time).

2. After the call to $BF06 the date and time locations should be nonzero and the MLI will use these values. If not the MLI will use the values that are already in the file's modified field. If you are making a new file the value will be zero.

If you have a Thunderclock, ProDOS will read the date and time automatically and a JMP to the routines starting address will be stored in $BF06. To use another card you will need a routine that reads your clock card then places the information in bytes $BF90 through $BF93. Use this format:

```
                $BF91                $BF90
           +7+6+5+4+3+2+1+0+    +7+6+5+4+3+2+1+0+
Date:      |     year     | month |     day     |
           +-+-+-+-+-+-+-+-+    +-+-+-+-+-+-+-+-+

                $BF93                $BF02
           +7+6+5+4+3+2+1+0+    +7+6+5+4+3+2+1+0+
Time:      |     hour     |      |    minute   |
           +-+-+-+-+-+-+-+-+    +-+-+-+-+-+-+-+-+
```

Your routine must begin with a CLD instruction and must end with a RTS. Don't forget to put your starting address in $BF06. Apple recommend that you put your routine in page three ($300-$3CF).

The following program works with the MBI clock card installed in slot 5. If you have a MBI card in another slot change line 4. If you are using a card other than MBI see your cards reference manual for a routine for your card. Load this program with your STARTUP program so you will be off and running without loading by hand. I would like to thank Bob Garth for all his help with this program. If any of you come across any routines for other clock cards please send them to me and I will see that the club library gets them. This program will be on the June DOM.

ProDOS & MBI clock card

```
 1          LST  OFF
 2          ORG  $0300
 3
 4 SLOT10   =    $50
 5 PAC      =    $C081+SLOT10
 6 PA       =    $C080+SLOT10
 7 PBC      =    $C083+SLOT10
 8 PB       =    $C082+SLOT10
 9
10 HOLD     =    $01
11 READ     =    $02
12 WRITE    =    $03
13 SEC30    =    $04
14
15
16 YEAR     =    $BFP1
17 MON      =    $BF90
18 HOUR     =    $BF93
19 MIN      =    $BF92
20
21
22          LDA  #/RTIME
23          STA  $BF0B
24          LDA  #RTIME
25          STA  $BF07
26 RTIME
27          CLD
28          LDA  #$00
29          STA  PAC
30          STA  PBC
31          LDA  #$80
32          STA  PA
33          LDA  #$FF
34          STA  PB
35          LDA  #$04
36          STA  PAC
37          STA  PBC
38          LDA  HOLD
39          STA  PB
40          LDA  #$04
41          JSR  WAIT
42          LDX  #$0C
43 LOOP     TXA
44          ASL
45          ASL
46          ASL
47          ASL
48          ORA  #READ+HOLD
49          STA  PB
50          LDA  PA
51          AND  #$0F
52          STA  TIME,X
53          DEX
54          BPL  LOOP
55          LDA  #$00
56          STA  PB
57          LDA  TIME+5
58          AND  #$03
59          STA  TIME+5
60          LDY  #$0C
61          JSR  CONVERT
62          STA  YEAR
63          LDA  #$0A
64          JSR  CONVERT
65          STA  MON
66          LDY  #$08
67          JSR  CONVERT
68          STA  DAY
69          LDY  #$05
70          JSR  CONVERT
71          STA  HOUR
72          LDY  #$03
73          JSR  CONVERT
74          STA  MIN
75          LDA  MON
76          ASL
77          ASL
78          ASL
79          ASL
80          ASL
81          STA  MON
82          LDA  YEAR
83          ROL
84          STA  YEAR
85          LDA  DAY
86          AND  #%00011111
87          ORA  MON
88          STA  MON
89          RTS
90
91 WAIT     SEC
92 LP0      PHA
93 LP1      SBC  #$01
94          BNE  LP1
95          PLA
96          SBC  #$01
97          BNE  LP0
98          RTS
99
100 CONVERT
101         LDA  TIME,Y
102         TAX
103         LDA  TBL,X
104         DEY
105         CLC
106         ADC  TIME,Y
107         RTS
108
109 TBL     DFB  0,10,20,30,
              40,50,60,70,80,90
110 DAY     DFB  0
111
112 TIME    DS   $D
```

Stories from the past...SEED Oct. 1985

GREEN APPLES

When it comes to math on a computer, we all have to re-learn certain procedures. To multiply, we use "*" instead of "x" and to divide we use a slash "/" because we don't have a "÷". (If we don't have a "÷", then how did he type one?)

One of the questions that comes up a lot is how to control the number of places to the right of a decimal you will work with. In other words, if you multiply 10.55 * .55 you would get 5.8025. What if you just wanted two places (5.80)? There is an easy way to solve this problem in BASIC using the INT function. The INT in BASIC is used to chop off all integers to samller than 1 or more - precisely the numbers to the right of the decimal point. To do this, you would type in the following lines of BASIC code:

```
100  let x = 5.8025
110  let y = INT(x)
```

In this example x would equal 5.

Well, that's all fine and dandy but we also lost the two places to the right of the decimal that we originally wanted (used a lot when computing dollars and cents). To make this work, we need to move the decimal point to the right by two places before we take the ineger value, then move it back again. The following example will do this nicely:

```
100 Let x = 5.8025
105 Let x = x * 100
110 Let y = INT x
115 Let y = y/100
```

Line 105 will change x to equal 580.25. Line 110 will set y to equal 580, and Line 115 will change y to equal 5.80. Ta Da!

If you wish to allow only one place to the right of the decimal, just multiply by 10 and divide by 10. If you wish to allow 3 places, multiply and divide by 1000, and so on.

For the more advanced mathematicians, you may have noticed problems with the precision of some of your math computations. For example, try this:

```
100 Let x = Z * 7
110 Let y = 7 ^ Z
120 Print "x = "; x
130 Print "y = "; y
```

Any mathematician would expect to have x and y produce the same answer. Wrongo! On the Apple, because of the way BASIC deals with flounting point math, you can run into some strange precision problems. If you tried this little program, you found that 7 times 7 equalled 49 but 7^2 equalled 49.0000001. This little bug has driven several programmers off the deep end. You can get around this problem using the INT function.

```
100 x = INT (7 ^ Z)
```

Sure, it's a dirty trick for your computer to play on you, but at least you can beat it in the end.

Stories from the past...SEED Aug. 1985

Dear John

JOHNNY APPLESEED

Dear John,

Often times, I use a GET statement in Applesoft to hold the screen, enabling the user to continue when he is ready. Unfortunately, using the GET statement can become a problem when I go to access the disk later in the program. Is there an alternate GET statement that won't interfere with disk operation??

GETaway

Dear GETaway,

Yes, you can replace your screen-holding get statement with Call 64795.

John

Dear John,

I am writing a program in Applesoft under Prodos that I would like to use the Open-Apple and Closed-Apple keys in. How does a program tell if one of the Apple keys is being pressed??

Opened And Closed

Dear Opened,

On the Apple //e and //c, the open Apple key is the same as paddle button #0 and the closed Apple key is the same as paddle button #1. Therefore, you can determine if they are being pressed by PEEKing the location of the paddle buttons, which is -16287 for paddle button #0, and -16286 for paddle button #1. This makes the complete statement to determine if the open-apple key is being pressed:

```
IF PEEK(-16287) > 127 THEN ?"It is now pressed"
```

Good luck!

John

Johnny Appleseed
c/o Denver Apple Pi
P.O.Box 17467
Denver, CO 80217-0467

Stories from the past...SEED Feb. 1986

Green Apples

by Randall Ansley

The LIST command is a very important part of programming in BASIC. It lets you view and edit your programs. LIST will also let others see what you have done or how you did it and that is not always desirable. Sometimes you just want to keep others from being able to list out your BASIC programs. If they can't LIST your program then they can't change it or even see how you go about doing what you do. Some people use BASIC compilers to protect their code but that means you have to go purchase a compiler.

There are many ways to make a BASIC program un-LIST-able. Some are very easy and others take some doing. Here are a few tricks to help you stop people from peeking at your code.

RUN SPOT RUN...

One way to stop LIST, HOME, and any other command that you can enter from the keyboard is to turn your Apple into a RUN ONLY mode. That way no matter what they type RUN will be what happens. To do this all you need to do is change one Byte in zero page. Woops! Looks like I just lost most of you. Let me put it another way...all you need to do is one POKE. Type in this line of code:

```
0  POKE 214,128
```

This will make all keyboard commands RUN you program. To go back to normal just type:

```
POKE 214,0
```

The down side of doing this is that your program must be run for the poke to take place. If someone does a LOAD then LIST they will be able to list your program because the poke never was executed. You may also run into some problems if your program calls on DOS to do anything, you could end up re-running the program. For simple LIST control this works jJust fine.

@!$*/@\\#~*%?/*!...

Another cute little trick is to let them list your program but don't let them see anything they can read. In other words turn the listing into garbage. You can do this with another poke. Type:

```
POKE 243,x
```

Let x be any number except 0, 32, 127, 160. Any other number will give you real strange stuff when you list. If you put in 0 or 127 the listing will be normal. If you put in 32 or 127 the listing will all be lower case but it can be read. This poke will not effect the normal running of your program most of the time but it can mess up your print statements so find a number that you like and play with it.

As with the first POKE this one will also only work if the program has been run.

A Booby Trap for real Boobies...

Now for a real fix that will keep them guessing. This little booby trap will make it hard for even the people that know what you did. This is not an easy fix like the others but if you follow the steps, I promise it will work for you.

This process will take two steps one in BASIC and one in the computers monitor. The first thing you want to do to your program is ad a line at the beginning of your program. Type in the following line of BASIC:

```
0 REMXXXXXXXSORRY NO PEEKINGXXXXX
```

The "X"s must be typed in as they are spaces we will use in the computers monitor. Type REM the 7 X's the message then 5 more X's. Now you must go to the monitor. Type :

```
CALL -151
```

This will put you in the monitor and give you the "*" prompt. Now you need to list the code that makes up the beginning of you BASIC program. The starting location of your progran is $801. The "$" tells you that this is a HEX. address but don't worry about that now.

To show the beginning of your program type:

801L

This will list one screen of memory starting with address 801. You will get a bunch of strange numbers and letters that look something like this:

```
*801L

0801-   23              ???
0802-   08              PHP
0803-   00              BRK
0804-   00              BRK
0805-   B2              ???
0806-   58              CLI
0807-   58              CLI
0808-   58              CLI
0809-   58              CLI
080A-   58              CLI
080B-   58              CLI
080C-   58              CLI
080D-   53              ???
080E-   4F              ???
080F-   52              ???
0810-   52              ???
0811-   59 20 4E        EOR   $4E20,Y
0814-   4F              ???
0815-   20 50 45        JSR   $4550
0818-   45 4B           EOR   $4B
```

Don't worry about what all this stuff is. If you want to know more about HEX and machine language you may wish to learn Assembly Language. For now we just want to change a few bytes in memory. Notice that at locations $806 - $80C we have a bunch of 58s on the right. These are HEX. bytes for the "X"s that we put in our BASIC REM statement. We are going to change these seven Xs into ^Hs (Cont. Hs). A ^H is the same as a backspace. To do this type the following line:

806: 08 08 08 08 08 08 08

Don't forget the colon and make sure that you put a space between each 08. If you re-list that section of code by typing 801L again you will see that the range from $0806 - $080C now has 08s on the right of each location number. What this change will do is hide the line number and the word REM when the program is listed. All the viewer will see is:

SORRY NO PEEKING

Now for the last change we need to list the next screen of memory. To do this just type:

L <RETURN>

This will list memory starting with about location *081A. At locations $081D - $0821 you will see more 58s. these are the five Xs that we typed after the no peeking message. These bytes will be used to put a FP command into the computer that will erase memory if someone tries to list you program. To put this little routine in type:

81D: 0D 04 46 50 0D

Without going into details this this booby trap will toss out your program if anyone tries to list it. The only thing they will see is the no peeking message the first time they try to list it. After that they will get nothing.

To get back to BASIC type:

CONTROL-C <RETURN>

Now save what you have done and then try to list it. The one disadvantage to this booby trap is that the smarter boobies may figure out that they can step over your trap by listing past it. Example: LIST 10- . To stop them in their tracks just repeat the steps in a few more places. If you do this you need to list through the monitor and look for the string of 58s and change them in the same way.

Hope these little tricks help you deal with your snoopers.

# # #

PERSONAL DOMAIN SOFTWARE

In addition to the collection of public domain software, the club has several personal domain and/or sales disks available. What do we mean by Personal Domain? Well, that means you are permitted to copy a copyrighted, commercial program so that you can try it to see if you find it worthwhile for your needs. If you want to continue using it after the trial period, the author, not Apple Pi, requests that you buy the program by sending a payment directly to the him or her. Usually the personal domain disk has abbreviated instructions on the disk and usually you receive an updated disk and/or printed manual if you purchase the program. Apple Pi does not endorse any commercial product nor does it seek to guarantee, enforce, or communicate with the author in any way. The list below includes the information on requested payment as it is found on the specific disk. If you copy any of these disks do not, I repeat, do not send money to Apple Pi!

Also, Apple Pi has several disks that are sold at $3.00 and $5.00. These disks offer exceptional value to you and address a specific topic or function. These disks are included first in the list below.

1. Universal Disk Labeler. $5.00/Sales Disk. This disk reads the catalog from a disk and writes a condensed version on a 1-7/16 x 4 inch label for application to the disk. Printers supported are Epson, C.Itoh, Nec, and Okidata. You can likely modify control codes for any printer, however.

2. Applesoft Tutor Disk. $5.00/Sales Disk. A superb tutorial on the applesoft language for the Apple series computer.

3. Epson Printer Disk. $5.00/Sales Disk. A collection of Epson printer utilities for setup, graphics dumps, and demonstration programs.

4. Imagewriter Printer Disk. $5.00/Sales Disk. A collection of programs for the Imagewriter Printer. More extensive than the Epson disk. Includes 21 fonts, a font editor, and download utility. A superb value.

5. Freewriter. $3.00/Sales Disk. A Prodos based modern word processor good as a first word processor to learn. Also available in the library for copy to your disk for $1.00.

6. Pascal Disk Labeler. $5.00/Sales Disk. Another disk label program supporting various printers but designed to catalog Pascal formated disks from the DOS 3.3 operating system.

7. EVE. $30.00/Personal Domain Disk. A terminal program for use with Hayes Micromodem II/IIe. Includes Xmodem data transfers and will upload/download binary and basic files also. Does not support macro's or a dialing directory.

8. Elfware Presents. $17.50/Personal Domain Disk. Two sided disk with coloring programs for young children. Also includes a pictorial-musical alphabet learning program.

9. Diversi-DOS. $30.00/Personal Domain Disk. Apple Pi charges $5.00 in addition for the copy on an Apple Pi disk. Will greatly speed up disk operations. Can move DOS to language card, includes keyboard buffer, keyboard macros, a type command, and improved Applesoft Editor.

10. One Key DOS. $9.00/Personal Domain Disk. Apple Pi charges $5.00 in addition for the copy on an Apple Pi disk. Modifies normal DOS 3.3 so that commands use only one keystroke.

11. Disk Manager. $29.00/Personal Domain Disk. A utility disk that remains language card resident. Provides features similar to Copy II+ and also a catalog arranger and diskette labeler.

12. Class Records. $30.00/Personal Domain Disk. Contains a series of programs designed for teachers to use in administering grades and other tasks.

13. Master Chart. $18.00/personal Domain Disk. The best of all the disks. Includes bar,line, and pie chart graphics routines and many other utilities. Will edit, display, and assemble shape tables. Very good as a graphic utility.

14. Diversicopy. $30.00/Personal Domain Disk. A fast copy utility for un-protected disks. Works best with a ram card. Will format Prodos, DOS, CPM, and Pascal blank disks.

15. Font Machine. $15.00/Personal Domain Disk. Is a utility to download 20 included fonts to parallel printers including Apple DMP, Epson FX-80, C.Itoh 8510 and similar printers with graphics capability. Does not include a font editor but Imagewriter Disk font editor should work.

If you are aware of additional Personal Domain Disks, please let me know. I think this marketing arrangement offers strong advantages to author and user and I'd like to build on our collection.

---Ed Thompson, Software Librarian---

ORDER FORM

Denver Apple Pi
P.O. Box 17467
Denver, Colorado 80217

ship to: _____

_____

Category	Sub-Category	Circle disk numbers desired from list below.
Bus. & Database	Business	111 112 113 114 115 116 117 118 119 120 131 151 152 153
	Business Mgmt	132 133
	Database	201 202 203 204 205 206
	Engineering	301 302 311 331
	Science	351 352 353 354 355 356 357 371 372 373 374 375 376 377
Language & Tutor	Computer Lang	401 402 403 404 405 406 407 408 409 410 411
	Computer Tutor	451 452 453 454 455 456 461 462 463 464 465 466 467 468
Music	Music	471 472 473 474 475 476 477 478
Education	Foreign Lang	501 502 503 504
	Math Programs	511 512 513 514 515 516 517 518 519 525 526 527 551 552 553 554 555
	Reading & Lang	541 542 543 544
	Science	561 562 563 564 565 571 572
	Education	581 582 583 584 585 586 587 588 589 590 591 592 593 594
Personal & Home	Home	601 602 621 622 631 632 641 642 643 644 645 646 647
Utilities	Utilities	675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
Eamon Games	Eamon	700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 763 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799
Games	Games	801 802 803 811 812 813 814 815 816 821 822 823 827 828 829 830 831 832 837 838 839 840 841 847 848 849 850 851 852 853 861 862 863 864 870 871 872 873 874 875 876 877 878 879 880 881 891 892 893 894 895 896 897 898
X-Rated	X-Rated	891 892 893 894
Communication	Hobby/Radio	901 902 903 911 912 913 914 915
Graphics & Shapes	Graphics	931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976
Inter. Apple Core	IAC	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 45 46 47 48 53 54 56 57 58 41 45 46 47 48 53 54 56 57 58
Disk of the Month	DOM	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
Personal Domain	Personal Domain	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Appleworks	Appleworks	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 60 61
Prodos	Prodos	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Pascal	Pascal	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
CP/M	CP/M	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198

There is a 10 copy side/member/day limit. Disks-of-the-month for the most recent 12 months are purchased for $5.00 each and are not copied. All other copies are $1.00/side or $2.00/side wirh disks.

Total DOM'S _____ x $5.00 = _____

Total Copies _____ x $1.00 = _____

Total Disks _____ x $1.00 = _____

Total Cost ..... _____

Note: Copies can be made to both sides of a disk. Also, there is no copy charge when a program is submitted with each copy request!

NEXT MEETING IS FEB. 5th

BRING A FRIEND AND GET IN FREE!

(IT'S FREE ANYWAY... PASS IT ON)

SEE YOU AT THE MEETING

THE SEED

denver apple pi

P.O. Box 17467
Denver, CO 80217

BULK RATE
U.S. POSTAGE
PAID
DENVER, CO
PERMIT-3182